Laplacian Eigenmaps

Máster Universitario en Ciencia de Datos - Métodos Funcionales en Aprendizaje Automático

Ángela Fernández Pascual

Escuela Politécnica Superior
Universidad Autónoma de Madrid

Academic Year 2021/22

Universidad Autónoma de Madrid

Contents

Laplacian Eigenmaps Algorithm

Laplacian Eigenmaps

- Spectral Dimensionality Reduction method.
- Proposed by Belkin and Niyogi in 2002.
- Aim: to reduce dimensionality for semi-supervised learning while preserving the local information.
- Special interest: data that lies in a smooth, compact, $\bar{M}$-dimensional manifold $\mathscr{M}$ embedded in the original space $\mathscr{M} \subset \mathbb{R}^M$.

Abstract

“One of the central problems in machine learning and pattern recognition is to develop appropriate representations for complex data. We consider the problem of constructing a representation for data lying on a low-dimensional manifold embedded in a high-dimensional space.

Drawing on the correspondence between the graph Laplacian, the Laplace Beltrami operator on the manifold, and the connections to the heat equation, we propose a geometrically motivated algorithm for representing the high-dimensional data. The algorithm provides a computationally efficient approach to nonlinear dimensionality reduction that has locality-preserving properties and a natural connection to clustering.

Some potential applications and illustrative examples are discussed.”

How it works?

Dimensionality Reduction Problem

Sample data: $\mathscr{S} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$ with $\mathbf{x}^{(i)} \in \mathbb{R}^{M}$.
New representation: $\{\hat{\mathbf{x}}^{(1)}, \dots, \hat{\mathbf{x}}^{(N)}\}$ with $\hat{\mathbf{x}}^{(i)} \in \mathbb{R}^{\bar{M}}$, where $\bar{M} \ll M$.

1. To organize $\mathscr{S}$ as a weighted graph of N nodes, defining edges only between neighbours.
2. To chose the weights (using the Laplacian matrix).
3. To compute the eigenvectors of this matrix for obtaining the embedded map.

LE - Step 1: Constructing the adjacency graph

Adjacency graph

$$\mathscr{G} = (\mathscr{V}, \mathscr{E}),$$

- $\mathscr{V}$: vertices of the graph,
- $\mathscr{E}$: edges of the graph.

We assume it is an **undirected weighted** graph.

- Defining our graph:
  - $\mathscr{V} = \{\mathbf{x}^{(i)}\}$.
  - $(i, j) \in \mathscr{E}$ if $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are near.
- How to define the neighbourhood?
  - ϵ-neighbourhood graphs
  - KNN graphs
  - Fully connected graphs

LE - Step 2: Choosing the weights

The previous graph will be defined always as a **weighted graph**, with weights w_{ij}.
Options for defining these weights:

1. **Simple-minded** method

$$\boxed{w_{ij} = \begin{cases} 1 & \text{if } (i,j) \text{ are connected,} \\ 0 & \text{if } (i,j) \text{ are not connected.} \end{cases}}$$

Advantage	we do not have to fix any parameter.
Disadvantage	it gives us less information about the local structure of the data.

2. **Heat Kernel** method with parameter $t \in \mathbb{R}$.

$$\boxed{w_{ij} = \begin{cases} e^{-\frac{\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\|^2}{4t}} & \text{if } (i,j) \text{ are connected,} \\ 0 & \text{if } (i,j) \text{ are not connected.} \end{cases}}$$

Advantage we can expect to obtain more information about the relationships inside the data set.

And with these weights we define a similarity matrix $\mathbf{W}$.

Notebook

LE: Steps 1 and 2

Step 3: Computing eigenmaps - L Definition

Degree Matrix

We define the **Degree Matrix** $\mathbf{D}$ of the similarity matrix $\mathbf{W}$ as the diagonal matrix with entries

$$d_i = \sum_{j=1}^{N} w_{ij}.$$

Unnormalized Graph Laplacian

We define the **Unnormalized Graph Laplacian** $\mathbf{L}$ as

$$\mathbf{L} = \mathbf{D} - \mathbf{W}.$$

Step 3: Computing eigenmaps - $\mathbf{L}$ Properties

Lemma ($\mathbf{L}$ Positive Semidefinite Matrix)

The unnormalized graph Laplacian $\mathbf{L}$ *is a positive semidefinite matrix, i.e.,* $f^\top \mathbf{L} f \geqslant 0 \quad \forall f$.

Proof.

$$\begin{aligned} f^\top \mathbf{L} f &= f^\top \mathbf{D} f - f^\top \mathbf{W} f = \sum_{i=1}^{N} d_i f_i^2 - \sum_{i,j=1}^{N} f_i f_j w_{ij} \\ &= \frac{1}{2}\left(\sum_{i=1}^{N} d_i f_i^2 - 2 \sum_{i,j=1}^{N} w_{ij} f_i f_j + \sum_{j=1}^{N} d_j f_j^2 \right) \\ &= \frac{1}{2}\left(\sum_{i,j=1}^{N} w_{ij} f_i^2 - 2 \sum_{i,j=1}^{N} w_{ij} f_i f_j + \sum_{i,j=1}^{N} w_{ji} f_j^2 \right) \\ &= \frac{1}{2} \sum_{i,j=1}^{N} w_{ij} (f_i - f_j)^2 \geqslant 0. \end{aligned}$$

□

Step 3: Computing eigenmaps

How can be the eigenmap computed?

- Defining the generalized spectral problem $\mathbf{L}\vartheta = \lambda \mathbf{D}\vartheta$.
- Computing the eigenvalues λ_i and eigenfunctions ϑ_i of $\mathbf{L}$.
  - $\mathbf{L}$ is a positive semidefinite matrix $\Rightarrow \lambda_i \geqslant 0$.
- Embedding in a $\bar{M}$-dimensional space: first $\bar{M}$ eigenvectors, without taking into account ϑ_0:

$$\boxed{\vartheta : \mathbf{x}^{(i)} \to \left(\vartheta_1(\mathbf{x}^{(i)}), \dots, \vartheta_{\bar{M}}(\mathbf{x}^{(i)})\right)}.$$

Notebook

LE: Step 3

Step 3: Computing eigenmaps - $\mathbf{L}$ Properties

λ_0 of $\mathbf{L}$

For matrix $\mathbf{L}$, $\lambda_0 = 0$ is always a trivial eigenvalue.

Proof.

Thanks to the normalization, our matrix satisfies:

$$\begin{pmatrix} d_1 - w_{11} & \dots & -w_{1n} \\ \vdots & & \\ -w_{n1} & \dots & d_n - w_{nn} \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 0 & \dots & 0 \end{pmatrix}.$$

□

- The trivial solution associated to $\lambda_0 = 0$ is not considered:
  - $\vartheta_0 : \mathbf{x}^{(i)} \to (1, \dots, 1)$ collapses all the elements of each point onto 1.
  - It gives a projection with a minimum distance between points but we lose all information.

Notebook

LE: Trivial Solution

LE: *Scikit-learn* implementation

```
class sklearn.manifold.SpectralEmbedding(n_components=2, *, affinity='nearest_neighbors',
        gamma=None, random_state=None, eigen_solver=None, n_neighbors=None, n_jobs=None)
```

n_components	Number of components of the embedding (reduced dimension) in Step 3.
affinity	How to construct the affinity matrix.

- 'nearest_neighbors' (by default).

  → n_neighbors: number of neighbours selected (by default $\max(\text{n\_samples}/10, 1)$).
- 'rbf': Radial Basis Function Kernel.

  → gamma: kernel coefficient (by default $1/\text{n\_features}$).

Notebook

LE: The Swiss Roll Example

Method Justification

Why does LE preserve local information?

Idea: The embedding $\mathbf{Y} : \mathscr{G} \to \mathbb{R}^{\bar{M}}$ minimizes a reconstruction error.

Assumptions: connected graph $\mathscr{G}$, and $\bar{M} = 1$ to simplify notation and explanations.

A **good map** will **minimize** the objective function defined by:

$$\boxed{\mathcal{J}(\mathbf{y}) = \frac{1}{2} \sum_{i,j} (y_i - y_j)^2 w_{ij}}.$$

Optimal Embedding (I)

Let's rewrite the previous optimal problem:

$$
\begin{aligned}
\frac{1}{2}\sum_{i,j}(y_i - y_j)^2 w_{ij} &= \frac{1}{2}\sum_{i,j}(y_i^2 + y_j^2 - 2y_iy_j)w_{ij} \\
&= \frac{1}{2}\left[\sum_i d_i y_i^2 + \sum_j d_j y_j^2 - 2\sum_{i,j} w_{ij}(y_iy_j)\right] \\
&= \frac{1}{2}\left[\mathbf{y}^\top \mathbf{D}\mathbf{y} + \mathbf{y}^\top \mathbf{D}\mathbf{y} - 2\mathbf{y}^\top \mathbf{W}\mathbf{y}\right] \\
&= \frac{1}{2}\left[2\mathbf{y}^\top \mathbf{D}\mathbf{y} - 2\mathbf{y}^\top \mathbf{W}\mathbf{y}\right] \\
&= \frac{1}{2}\left[2\mathbf{y}^\top (\mathbf{D} - \mathbf{W})\mathbf{y}\right] \\
&= \boxed{\mathbf{y}^\top \mathbf{L}\mathbf{y}}.
\end{aligned}
$$

Optimal Embedding (II)

1. To minimize $\mathcal{J}(\mathbf{y}) \equiv$ to solve the matrix minimization problem:

$$\arg\min_{\mathbf{y}} \{\mathbf{y}^\top \mathbf{L} \mathbf{y}\}$$
$$s.t.\ \mathbf{y}^\top \mathbf{D} \mathbf{y} = 1.$$

- In general, the restriction is defined as $\|\mathbf{y}\|^2 = 1$.
- In this case, **D** seems to be the natural measure over the graph:
  - Big values of $d_i \Rightarrow \mathbf{x}^{(i)}$ is very connected $\Rightarrow \mathbf{x}^{(i)}$ is more important.

2. Let's rewrite the problem for solving it using **Lagrange multipliers**:

$$\phi(\mathbf{y}) = \mathbf{y}^\top \mathbf{L} \mathbf{y} - \lambda(\mathbf{y}^\top \mathbf{D} \mathbf{y} - 1),$$
$$\nabla\phi(\mathbf{y}) = \mathbf{L}\mathbf{y} - \lambda \mathbf{D}\mathbf{y} = 0.$$

- Solution: eigenvector $\mathbf{y}$ corresponding to the minimum eigenvalue λ of $\mathbf{L}$, that satisfies $\mathbf{L}\mathbf{y} = \lambda \mathbf{D}\mathbf{y}$.

Optimal Embedding (III)

- Recall that the vector $\mathbf{y} = \mathbf{1} = (1, \ldots, 1)$ corresponds to an eigenvalue 0 (not interesting).
- Let's eliminate this possibility:

$$\arg\min_{\mathbf{y}} \{\mathbf{y}^\top \mathbf{L} \mathbf{y}\}$$

$$s.t. \begin{cases} \mathbf{y}^\top \mathbf{D} \mathbf{y} = 1 \\ \boxed{\mathbf{y}^\top \mathbf{D} \mathbf{1} = 0} \end{cases}.$$

Solution eigenvector corresponding to the smallest non-zero eigenvalue.

Conclusion LE $\equiv$ this minimization problem $\Rightarrow$ LE is a good method for preserving local information (when $\bar{M} = 1$).

Optimal Embedding (IV)- Generalization to $\bar{M}$-dimension

Embedding $\mathbf{Y} = [\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(\bar{M})}]$

Cost function $\mathcal{J}(\mathbf{Y}) = \sum_{ij} \|\mathbf{y}^{(i)} - \mathbf{y}^{(j)}\|^2 w_{ij}$

Equivalent problem $\mathrm{Tr}(\mathbf{Y}^\top \mathbf{L}\mathbf{Y})$

Proof.

$$\begin{aligned}
\sum_{ij} \|\mathbf{y}^{(i)} - \mathbf{y}^{(j)}\|^2 w_{ij} &= \sum_{ij} \|((y_1^{(i)} - y_1^{(j)}), \ldots, (y_{\bar{M}}^{(i)} - y_{\bar{M}}^{(j)}))\|^2 w_{ij} \\
&= \sum_{ij} \left((y_1^{(i)} - y_1^{(j)})^2 + \cdots + (y_{\bar{M}}^{(i)} - y_{\bar{M}}^{(j)})^2 \right) w_{ij} \\
&= \sum_{ij} (y_1^{(i)} - y_1^{(j)})^2 w_{ij} + \cdots + \sum_{ij} (y_{\bar{M}}^{(i)} - y_{\bar{M}}^{(j)})^2 w_{ij} \\
&= \mathbf{y}_1^\top \mathbf{L}\mathbf{y}_1 + \cdots + \mathbf{y}_{\bar{M}}^\top \mathbf{L}\mathbf{y}_{\bar{M}} \\
&= \mathrm{Tr}(\mathbf{Y}^\top \mathbf{L}\mathbf{Y}).
\end{aligned}$$

$\square$

Optimal Embedding (V)- Generalization to $\bar{M}$-dimension

- Problem to solve:

$$\arg\min_{\mathbf{Y}}\{Tr(\mathbf{Y}^\top \mathbf{L}\mathbf{Y})\}$$
$$s.t.\ \mathbf{Y}^\top \mathbf{D}\mathbf{Y} = 1.$$

- For avoiding a collapse onto a $\bar{M}$-dimensional subspace, we should also add orthogonality restrictions.

Solution eigenvector matrix corresponding to the lowest eigenvalues of the generalized spectral problem $\mathbf{L}\mathbf{Y} = \lambda \mathbf{D}\mathbf{Y}$.

Conclusion The general LE algorithm is a good method for embedding a sample preserving its local information.

The Laplace Beltrami Operator (I)

- **Laplace Beltrami operator** Δf on a manifold $\mathscr{M} \equiv$ Laplacian $\mathbf{L}$ of a graph.
- Definition: $\Delta f = -div\nabla(f)$.

- Map: $f : \mathscr{M} \rightarrow \mathbb{R}$, for sending nearby points in $\mathscr{M}$ to nearby points in $\mathbb{R}$.
  - f at least twice differentiable.
- Two neighbour points $\mathbf{x}, \mathbf{z} \in \mathscr{M}$.
- Let's study $|f(\mathbf{z}) - f(\mathbf{x})|$ in the new space...

Consider a **geodesic curve** $\mathcal{C}$ parametrized by length with origin in $\mathbf{x}$, i.e.,

$$\begin{aligned} r &= d_{\mathscr{M}}(\mathbf{x}, \mathbf{z}), \\ \mathbf{z} &= \mathcal{C}(r), \\ \mathbf{x} &= \mathcal{C}(0), \\ f(\mathcal{C}(t)) &= g(t). \end{aligned}$$

The Laplace Beltrami Operator (II)

- Since $f(\mathbf{x}) = f(\mathcal{C}(0)) = g(0)$ and $f(\mathbf{z}) = f(\mathcal{C}(r)) = g(r)$, we can rewrite the difference as:

$$\begin{aligned} f(\mathbf{z}) - f(\mathbf{x}) = g(r) - g(0) &= \int_0^r g'(t)dt \\ &= \int_0^r \nabla f(\mathcal{C}(t)) \cdot \mathcal{C}'(t)dt. \end{aligned}$$

- Taking absolute values and using the Schwarz inequality, we arrive at:

$$\begin{aligned} |f(\mathbf{z}) - f(\mathbf{x})| &\leqslant \int_0^r \|\nabla f(\mathcal{C}(t))\| \|\mathcal{C}'(t)\| dt \\ &= \int_0^r \|\nabla f(\mathcal{C}(t))\| dt = \int_0^r \|\nabla f(\mathbf{x})\| dt + o(r) \\ &\leqslant r\|\nabla f(\mathbf{x})\| + o(r) = \|\mathbf{z} - \mathbf{x}\| \|\nabla f(\mathbf{x})\| + o(\|\mathbf{z} - \mathbf{x}\|). \end{aligned}$$

$\Rightarrow$ $\|\nabla f\|$ provides us with an estimate of how far apart f maps nearby points.

The Laplace Beltrami Operator (III)

- **How to define the map that better preserves local information?**
- As usual, we have to solve the minimization problem in terms of the reconstruction error:

$$\min_f \left\{ \int_{\mathcal{M}} \|\nabla f(\mathbf{x})\|^2 \right\}$$
$$s.t \; \|f\|_{\mathbf{L}^2(\mathcal{M})} = 1.$$

Note $\|\nabla f(\mathbf{x})\|$ gives a measure of the distortion between nearby points introduced by f.

- $$\int_{\mathcal{M}} \|\nabla f\|^2 = \int_{\mathcal{M}} \Delta(f) f.$$

Note $\Delta f = -div \nabla(f)$.

The Laplace Beltrami Operator (IV)

- Applying the Gauss's Divergence Theorem:

$$\begin{aligned}\int_{\mathcal{M}} < \nabla f, \nabla f > &= -\int_{\mathcal{M}} div(\nabla(f))f \\ &= -\int_{\mathcal{M}} \Delta(f)f.\end{aligned}$$

- Equivalent minimization problem:

$$\min_f \left\{ -\int_{\mathcal{M}} \Delta(f)f \right\}$$
$$s.t \ \|f\|_{\Delta^2(\mathcal{M})} = 1.$$

  - Δ positive semidefinite operator $\Rightarrow$ a minimum of the problem is given by an eigenfunction of Δ.
- Optimal embedding: first $\bar{M}$ eigenfunctions corresponding to $0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{\bar{M}}$ of Δ:

$$\mathbf{x} \to (f_1(\mathbf{x}), \ldots, f_{\bar{M}}(\mathbf{x})).$$

The Laplace Beltrami Operator (V)

We have just argued that LE is not only a good method for embedding points in a graph in $\mathbb{R}^M$, but also works properly, without changing the algorithm proposed, if the sample data set is included in a **smooth manifold of a lower dimension** than the one of the original space.

The Heat Equation

- The Laplacian Operator is intimately related to the **heat flow**.
- The Heat equation is

$$\left(\frac{\partial}{\partial t} + L\right) u = 0,$$

  where
  - $u(\mathbf{x}, t)$ is the heat distribution in time t,
  - $f(\mathbf{x}) = u(\mathbf{x}, 0)$ is the initial heat distribution with $f : \mathcal{M} \to \mathbb{R}$.
- Its solution is given in terms of the Heat kernel H_t:

$$u(\mathbf{x}, t) = \int_{\mathcal{M}} H_t(\mathbf{x}, \mathbf{y}) f(\mathbf{y}).$$

  - $H_t(\mathbf{x}, \mathbf{y})$: how much heat flows from $\mathbf{y}$ to $\mathbf{x}$ in time t.

Objective To find the solution of the LE problem in terms of the Heat kernel.

Notebook

Heat flow

The Heat Kernel

- Let's rewrite the minimization problem $Lf(\mathbf{x})$ using the Heat Kernel:

$$Lf(\mathbf{x}) = Lu(\mathbf{x},0) = -\frac{\partial}{\partial t}u(\mathbf{x},t)\bigg|_{t=0}$$
$$= -\frac{\partial}{\partial t}\left[\int_{\mathcal{M}} H_t(\mathbf{x},\mathbf{y})f(\mathbf{y})\right]\bigg|_{t=0}.$$

- Assuming an exponential coordinate system, the Heat kernel is like a Gaussian function:

$$H_t(\mathbf{x},\mathbf{y}) = (4\pi t)^{-\frac{k}{2}} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{4t}}(\phi(\mathbf{x},\mathbf{y}) + O(t)),$$

where $\phi(\mathbf{x},\mathbf{y})$ is a smooth function with $\phi(\mathbf{x},\mathbf{x}) = 1$.

- And when $\mathbf{x}$ and $\mathbf{y}$ are very close and t is small:

$$\boxed{H_t(\mathbf{x},\mathbf{y}) \approx (4\pi t)^{-\frac{k}{2}} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{4t}}}.$$

Choosing the graph weights

- When $t \to 0$, the Heat Kernel is more localized and tends to Dirac's δ-function:

$$\lim_{t\to 0} \int_{\mathcal{M}} H_t(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) = f(\mathbf{x}).$$

- Applying the Heat Kernel equation, considering a small t:

$$\begin{aligned}
Lf(\mathbf{x}) = Lu(\mathbf{x}, t) &= -\frac{\partial}{\partial s} u(\mathbf{x}, s)\bigg|_{s=t} \\
&\cong -\lim_{t\to 0} \frac{u(\mathbf{x}, t) - u(\mathbf{x}, 0)}{t} \\
&\cong \frac{1}{t}\left(f(\mathbf{x}) - \int_{\mathcal{M}} H(\mathbf{x}, \mathbf{y}) f(\mathbf{y}) d\mathbf{y}\right) \\
&\cong \frac{1}{t}\left(f(\mathbf{x}) - (4\pi t)^{-\frac{k}{2}} \int_{\mathcal{M}} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{4t}} f(\mathbf{y}) d\mathbf{y}\right).
\end{aligned}$$

Choosing the graph weights

- If $\mathbf{x}_1, \ldots, \mathbf{x}_k$ are points in $\mathcal{M}$:

$$Lf(\mathbf{x}_i) \cong \frac{1}{t}\left(f(\mathbf{x}_i) - \frac{1}{k}(4\pi t)^{-\frac{k}{2}} \sum_{\substack{\mathbf{x}_j \\ 0<\|\mathbf{x}_i-\mathbf{x}_j\|<\epsilon}} e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{4t}} f(\mathbf{x}_j) \right).$$

- Writing $\alpha = \frac{1}{k}(4\pi t)^{-\frac{k}{2}}$, and considering $f = 1$, then $Lf = 0$:

$$\frac{1}{\alpha} = \sum_{\substack{\mathbf{x}_j \\ 0<\|\mathbf{x}_i-\mathbf{x}_j\|<\epsilon}} e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{4t}}$$

$$\Rightarrow \alpha = \left(\sum_{\substack{\mathbf{x}_j \\ 0<\|\mathbf{x}_i-\mathbf{x}_j\|<\epsilon}} e^{-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{4t}} \right)^{-1}.$$

$\Rightarrow$ The weights are normalized and they sum $\frac{1}{t}$.

Choosing the graph weights

Laplacian expression:

$$Lf(\mathbf{x}_i) = d_i f(\mathbf{x}_i) - \sum_j w_{ij} f(\mathbf{x}_j),$$

Comparing with the Heat Kernel expression ($\alpha = 1$), the weights of the Laplacian Graph must take the values:

$$\boxed{w_{ij} = \begin{cases} e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{4t}} & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\| < \epsilon, \\ 0 & \text{otherwise.} \end{cases}}$$

LE Advantages and Disadvantages

LE Advantages and Disadvantages

Advantages

- LE lets us interpret our data in a geometric way.
- It is insensitive to outliers and noise.
- It exhibits stability with respect to the embedding, because this approach is based on the intrinsic geometric structure of the embedded manifold.
- It is simple to implement as it basically requires to solve an eigenvalue problem.

Disadvantages

- For a new sample point, we should repeat the whole algorithm over the new complete sample to reduce its dimension.
- It is difficult to select values for the parameters $\bar{M}$, the reduced dimension, and t, the Heat Kernel parameter as they are data-dependent.
- The approximation presented only handles manifolds from which data is sampled uniformly, but this rarely happens in real applications.

LLE from a Laplacian perspective

LLE from a Laplacian perspective

- LLE can be also seen from a Laplacian point of view.
- This perspective change just affect to Step 3, when obtaining the embedding coordinates.

- The key idea is that to obtain an **embedding** of the original points following the LLE algorithm we can just look for the **eigenvectors of $\mathbf{L}^2$**, that coincide with those of $\mathbf{L}$.

LLE from a Laplacian perspective

Lemma

$$\mathbf{M}f \approx \frac{1}{2}\mathbf{L}^2 f$$

Proof.

1. For a fixed point $\mathbf{x}^{(i)}$, and with $\mathbf{H}$ the Hessian matrix of f:

$$[(\mathbf{I}-\mathbf{W})f]_i \approx -\frac{1}{2}\sum_j m_{ij}(\mathbf{x}^{(i)} - \mathbf{x_i}^{(j)})^\top \mathbf{H}^{(i)}(\mathbf{x}^{(i)} - \mathbf{x_i}^{(j)}).$$

2. Defining $\mathbf{v}^{(j)} = \mathbf{x_i}^{(j)} - \mathbf{x}^{(i)}$ and assuming that $\sqrt{w_{ij}}\mathbf{v}_i$ form an orthonormal basis:

$$\mathrm{E}(\mathbf{v}^\top \mathbf{H}\mathbf{v}) = r\mathbf{L}f,$$

   where $r = \mathrm{E}(< \mathbf{v}^{(i)}, e_i >^2)$, and e_i form an orthonormal basis for the Hessian matrix $\mathbf{H}$.

3. Putting together 1 and 2: $(\mathbf{I}-\mathbf{W})^\top(\mathbf{I}-\mathbf{W})f \approx \frac{1}{2}\mathbf{L}^2 f$.

□

References

References

- Belkin, M., Niyogi, P. (2001, December). **Laplacian Eigenmaps and spectral techniques for embedding and clustering**. In Nips (Vol. 14, No. 14, pp. 585-591).
- Belkin, M., Niyogi, P. (2003). **Laplacian Eigenmaps for dimensionality reduction and data representation**. Neural computation, 15(6), 1373-1396.
- Belkin, M., Niyogi, P. (2007). **Convergence of Laplacian Eigenmaps**. Advances in Neural Information Processing Systems, 19, 129.